

## An New Load Allocation Algorithms of Direct Load Control

김 정 옥\*  
(Jeong-Uk Kim)

**Abstract** - This paper presents an advanced load allocation algorithm in Direct Load Control(DLC) system. It is important to aggregate a various demand side resource which is surely controllable at the peak power time for a successful DLC system. Previous load allocation algorithm appropriate for DLC system is based on interchanged information, but, this algorithm can not derive optimal solutions. In this paper, we develop the optimal algorithm and the new load allocation algorithm in polynomial time. The simulation results show that the proposed heuristic algorithm for DLC system is very effective.

**Key Words** : DLC(Direct Load Control), Peak control, Load control

### 1. 서 론

직접부하제어시스템은 냉난방 부하의 증가 등으로 인하여 전력수급에 문제가 발생할 소지가 있을 때, 수용가의 부하 설비를 직접 제어하여 안정적으로 전력을 공급하는 시스템이다. 2001년부터 한전과 에너지관리공단에서 해당 사업을 추진하였다. 직접부하제어사업은 전력계통의 비상수급상황에서 제어 가능한 부하자원을 확보함으로써 전력수급 안정화에 기여할 뿐만 아니라, 보조서비스(Ancillary Service) 시장의 참여를 통해서 전력설비투자를 지연할 수 있는 대안으로서 인식되어왔다.[1][2] 최근, 고유가에 의한 전력생산비 증가등으로 직접부하제어의 중요성이 더욱 부각되고 있다.

에너지관리공단(현재는 전력거래소에서 운영하고 있음)에서 시행하고 있는 직접부하제어 사업은 부하 사업에 참여하는 수용가를 대상으로 부하 제어 여부를 결정하고, 수용가의 제어 대상 설비의 전력 공급을 차단하는 방식을 사용하고 있다. 직접부하제어 시스템의 부하배분 알고리즘은 선정된 수용가의 부하 차단량의 합이 주어진 전체 부하 차단량과 같거나 크도록 수용가들을 선정하여야 한다. 기존에 제시되었던 부하배분 알고리즘은[3] 여러 가지 구현 방안을 제시하였지만, 최적해를 찾지는 못한다.

본 연구에서는 직접부하제어 시스템의 운영 방식 및 기존의 부하배분 알고리즘을 고찰하고, 주어진 시간 내에 최적의 해를 도출하기 위한 새로운 부하배분 알고리즘을 제시한 후에 시뮬레이션을 통하여 제시된 방법의 성능시험 결과를 보이도록 한다.

### 2. 본 론

#### 2.1 직접부하제어 시스템의 구성 및 기능

직접부하제어 시스템의 부하 제어 방식은 제어 지원금 산정과 밀접하다. 즉, 전일예고, 당일예고, 긴급예고 형태로 수용가측에 제어 알림이 이루어지며, 이에 대하여 수용가측의 제어 확인 응답이 이루어진 경우에 한하여 부하제어 가능 자원으로 등록되고, 부하제어 시점에 부하관리사업자시스템(Load Service Entity System; LSES)으로부터 직접부하제어장치(Energy Management Device; EMD)로 제어 신호가 전달되면 부하제어단말장치(Load Control Unit; LCU)에서 해당 설비의 전원차단에 의한 부하제어가 이루어진다.[4][5]

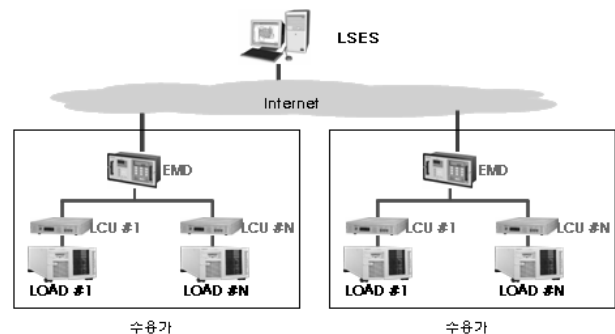


그림 1 직접부하제어 시스템  
Fig. 1 DLC System

#### 2.2 기존 부하배분 알고리즘

부하관리사업자시스템은 전력거래소의 부하제어할당량을

\* 정 회 원 : 호서대학교 벤처전문대학원 조교수

E-mail : jukim@hoseo.edu

접수일자 : 2009년 9월 26일

최종완료 : 2009년 12월 11일

기반으로 하여, 수용가의 계약용량 또는 사용량을 기반으로 부하를 배분하고, 인터넷을 이용하여 수용가와 실시간 협의를 수행한 후에, 부하제어를 실행하게 된다. 부하배분 알고리즘은 그림 2와 같은 절차에 의하여 수행된다.[3] 부하배분 알고리즘은 제어 Priority가 낮은 부하를 우선적으로 배분하는 방식과 제어 지원금이 낮은 부하를 우선적으로 배분하는 방식, 두가지를 혼합한 방식으로 크게 구분할 수 있다. 여러 방식 모두 수용가 부하를 특정 기준에 의하여 그룹화하여, 부하를 배분하므로 알고리즘의 적용에 있어서는 동일하다. 부하 배분 알고리즘 가운데 가장 대표적인 Load Shedding Priority Based Algorithm은 각 LCU의 제어 우선순위(priority)에 따라 부하제어량을 배분하는 것을 기본으로 한다. 만약, 제어 최종순위에 해당하는 LCU가 다수 존재할 경우에는 최종순위에 배분된 제어량의 크기에 가까운 차단가능량을 갖는 LCU부터 제어한다.

### 2.3 기존 부하제어 알고리즘의 문제점

배분 알고리즘은 비용을 최소화하는 해를 찾는 문제로서, 주어진 총 할당량 보다 크면서, 비용이 최소화되는 차단량의 집합이다. 기존 배분 알고리즘에서의 비용함수는 Priority 또는 제어지원금 수준에 대응하는 차단량으로, 수용가 부하별로 Priority와 제어지원금 수준에 관한 테이블을 유지하여야 한다. 기존의 배분 알고리즘은 최적해를 찾지 못한다.

예를 들어, RC=48이고, L1=40, L2=35, L3=30, L4=25 라고 하자. |RC-L1| = 8, |RC-L2| = 13, |RC-L3| = 18, |RC-L4| = 23 이므로, Norm이 가장 작은 L1을 선택하게 되어, RC=8이다. 두 번째 루프에서 |RC-L2| = 27, |RC-L3| = 22, |RC-L4| = 17 이므로, Norm이 가장 작은 L4를 선택하게 된다. RC=48인 경우에, L1과 L4를 선택하였으므로 L1+L4=65이다. 이 경우에 최적의 해는 L3+L4 = 30+25 =55이다.

### 3. 새로운 부하배분 알고리즘

새로운 부하배분 알고리즘은 에너지관리공단의 배분 알고리즘과 같이 수용가의 각 부하를 그룹핑한 후에, 전력 차단량을 넘지 않으면서 비용을 최소화하는 그룹을 선정하고, 남은 차단량이 특정 그룹의 차단량의 합보다 작은 경우에는 그룹내에서 부하 집합을 선정하도록 한다. 본 논문에서는 그룹내에서 부하를 선정하기 위하여 최적 알고리즘과 Heuristic 알고리즘을 제시한다.

부하배분 알고리즘은 비용을 최소화 하려는 목적을 가지므로, 목적함수를 다음과 같이 표시할 수 있다.

$$MIN \sum_i \sum_j f(a_{ij}) \times c_{ij}$$

여기서,  $a_{ij}$  = 수용가  $i$ 의 부하  $j$ 의 차단량,  $c_{ij} = 0$  또는 1 제약조건은 다음식으로 표현된다.

$$\sum_i \sum_j a_{ij} \times c_{ij} \geq B$$

여기서, B는 KPXB로부터의 전력 차단 할당량이고, 해는 {  $c_{ij} = 1$ 인  $a_{ij}$ 의 집합 } 이다.

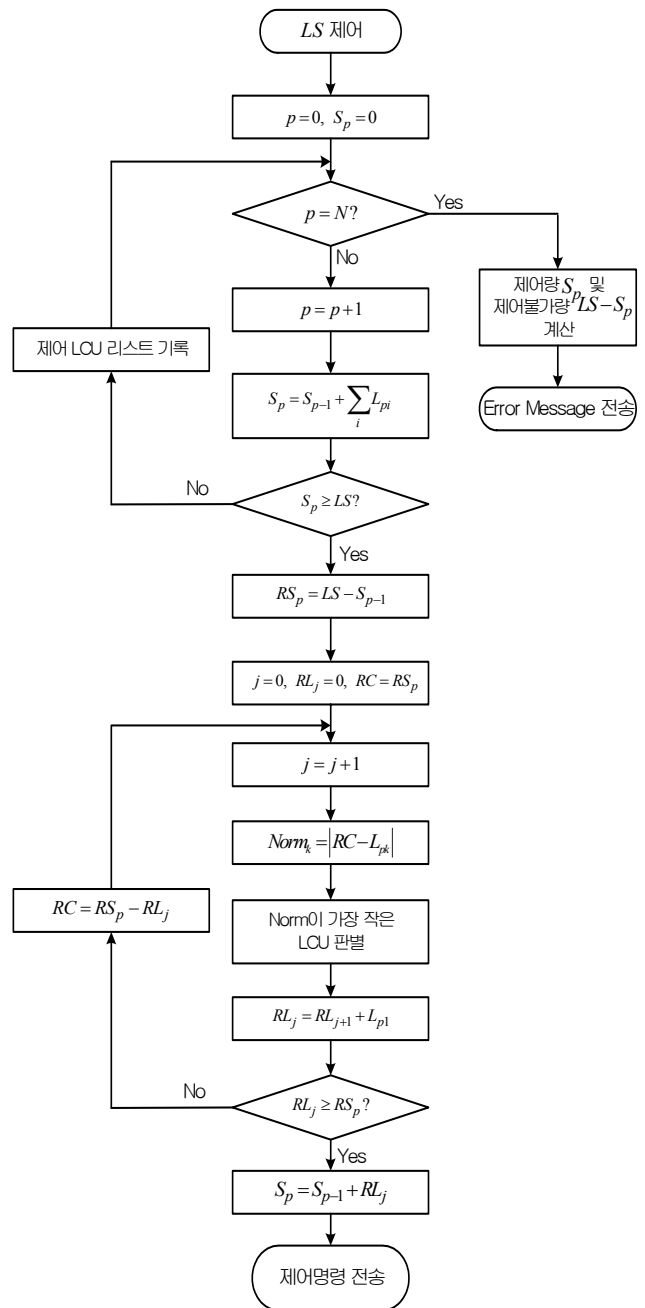


그림 2 부하배분 알고리즘  
Fig. 2 Load Allocation Algorithm

#### 3.1 최적 배분 알고리즘

최적 배분 알고리즘은 아래와 같이 Subset Sum 알고리즘을 활용할 수 있다. Subset Sum 알고리즘은 주어진 정수 집합과 정수 s에 대하여, 그 합이 s와 같은 값을 갖는 정수 집합을 찾는 문제이다. 고정 값 N에 대하여  $L_i$ 는  $2^i$ 에 비례한 수행 시간이 필요하므로 Exponential Time 알고리즘이다.

```

1.  $n \leftarrow |S|$ 
2.  $L_0 \leftarrow \langle 0 \rangle$ 
3. for  $i = 1$  to  $n$  do
4.    $L_i \leftarrow \text{MergeLists}(L_{i-1}, L_{i-1} + x_i)$ 
5.    $L_i$ 에서  $t$ 보다 큰 최소값보다 큰 수를 제거
6.  $L_n$ 에서 목표값보다 큰 최소값을 선정
    
```

그림 3 최적 배분 알고리즘

Fig. 3 Optimal Allocation Algorithm

최적 배분 알고리즘에서  $S=\{1, 4, 5\}$ 이고 목표값이 8이라면,  $L_0 = 0, L_1 = 0, 1, L_2 = 0, 1, 4, 5, L_3 = 0, 1, 4, 5, 6, 9, 10$ 이고, 목표값 8보다 큰 최소값은 9가 된다.

### 3.2 새로운 부하배분 알고리즘

Subset Sum 문제는 Exponential time이 필요하므로 NP-Complete 문제이다. 부하배분 알고리즘의 수행 시간을 감소하기 위하여 본 논문에서 제시하는 Polynomial 시간 알고리즘은 아래와 같다. 새로운 알고리즘은 일반적인 Approximation Subset 알고리즘을 수정하여 제시되었다. Subset Sum 문제는 목표값보다 적은 값을 찾지만, 부하배분에서는 목표값 이상의 값을 찾아야 하기 때문이다.

```

새로운 부하배분 알고리즘( $S, t, \delta$ )
1.  $n \leftarrow |S|$ 
2.  $L_0 \leftarrow \langle 0 \rangle$ 
3. for  $i = 1$  to  $n$  do
4.    $L_i \leftarrow \text{MergeLists}(L_{i-1}, L_{i-1} + x_i)$ 
5.    $L_i \leftarrow \text{Trim}(L_i, \delta)$ 
6.    $L_i$ 에서  $t$ 보다 큰 최소값을 초과하는 큰 수를 제거
7.  $L_n$ 에서 목표값보다 큰 최소값을 선정
    
```

그림 4 새로운 부하배분 알고리즘

Fig. 4 New Load Allocation Algorithm

최적해를 도출하면서, 수행 시간을 감소시키기 위하여 파라미터  $\delta$ 를 이용하여 리스트  $L_i$ 를 자르는 방법은  $\text{Trim}(L, \delta)$ 를 이용한다.

```

1.  $m \leftarrow |L|$ 
2.  $L' \leftarrow \langle 0 \rangle$ 
3.  $last \leftarrow Y_1$ 
4. for  $i = 2$  to  $n$  do
5.   if  $Y_i > last * (1+\delta)$  then
6.     append  $Y_{i-1}$  onto the end of  $L'$ 
7.      $last \leftarrow Y_i$ 
8. return  $L'$ 
    
```

그림 5 Trim( $L, \delta$ )

Fig. 5 Trim( $L, \delta$ )

만약, 리스트가 새로 만들어진 후에 리스트에 있는 여러 값들이 서로 근접해 있다면, 그 값들 중에 오직 한 값만 유지하도록 한다. 주어진 파라미터  $\delta$ 에 대해서 여기서  $0 < \delta < 1, y/(1+\delta) \leq z \leq y$ 라면, 값  $z$ 는  $y$ 에 근사한 값이다. 리스트  $L_i$ 의 갯수를 줄이기 위해서 가능한한 근사한 값을 많이 줄여나가는 방법이다. 예를 들어  $\delta = 0.1$ 이고,  $L =$

$\langle 10, 11, 12, 15, 20, 21, 22, 23, 24, 29 \rangle$  이라면, 근사값을 제거한 후에는  $L = \langle 10, 11, 15, 20, 23, 29 \rangle$ 이 된다. 12는 11의 근사값이고 21과 22는 20의 근사값이며, 24는 23의 근사값이기 때문이다. 알고리즘의 정확도와 수행시간은 Trade Off의 관계가 있다. 파라미터  $\delta$  값이 커질수록, 수행시간은 감소되지만, 최적해와의 오차가 커진다. 파라미터  $\delta$  값은 데이터의 패턴에 따라 결정되어야 하며,  $n$  값이 커질수록 큰  $\delta$  값을 설정하여야 한다. 직접부하배분에서는 알고리즘의 수행시간에 따라, 0.1~0.5 사이의 값을 활용할 수 있다.

### 3.3 Greedy 알고리즘

에너지관리공단에 구현된 기존의 부하배분 알고리즘은 Greedy 알고리즘의 일종으로, 다음과 같이 표현될 수 있다. Greedy 알고리즘은  $O(N^2)$  알고리즘이다.

```

1. while(TRUE)
2.    $Norm_j = |RC - X_j|$ 
3.    $Norm_j$ 가 가장 작은  $X_j$ 를 선택
4.    $RC = RC - X_j$ 
5.   if  $RC < 0$ , then return
    
```

그림 6 Greedy 알고리즘

Fig. 6 Greedy Algorithm

## 4. 시뮬레이션

부하 할당 문제는 NP-Complete 문제로서, 정해진 시간내에 일정한 성능을 만족하는 해를 찾기 위하여, 본 논문에서는 Polynomial 시간내에 수행되는 새로운 알고리즘을 제시하였다. 제안된 방법의 유효성을 검증하기 위하여 시뮬레이션을 수행하였다. 시뮬레이션에서는 직접부하제어 시스템에서 실제 활용되는 수용가 부하 데이터를 이용하여 두가지 방법으로 시뮬레이션을 수행하였다. 알고리즘은 Windows 기반하에 C++를 이용하여 구현되었다.

### 4.1 수용가수에 의한 시뮬레이션

첫 번째 시뮬레이션은 수용가수를 변경하면서 최적 배분 알고리즘의 수행시간을 측정하였다. 최적해 알고리즘의 수행시간이  $n$ 의 증가에 따라 기하급수적으로 증가함을 알 수 있다. 수용가 수 600개에 대하여 일정한 시간내(아마도 10분에서 1시간이 될 것이다)에 결과를 도출하기는 어려울 것으로 예상된다.

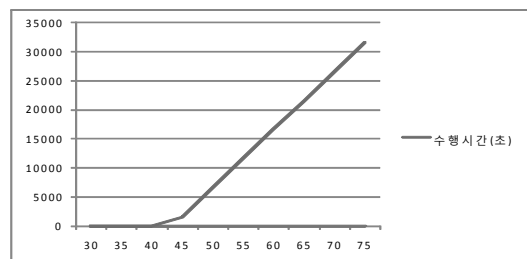


그림 7 최적 배분 알고리즘(수행시간)

Fig. 7 Optimal Allocation Algorithm(Time)

### 4.2 수행시간에 의한 시뮬레이션

직접부하제어는 차단시간 몇시간 전에 수용가로 차단 통보를 하는 경우가 있으므로, 알고리즘의 수행시간에 제약이 발생한다. 이 경우에 알고리즘은 수십 분 정도이내에 수행이 되어야 한다. 두 번째 시뮬레이션에서는 시간 제약(10분) 하에 알고리즘의 수행결과를 비교하였다. 최적해 알고리즘과 제안된 알고리즘을 제한된 시간동안만 수행하도록 알고리즘을 수정하였다. Greedy 알고리즘은 수행 시간이 짧아 시간 제약이 의미가 없다. 표 1에 Greedy 알고리즘과 최적 알고리즘, 제안 알고리즘의 결과를 비교하였다. 표 1에서 보는 바와 같이 제안 알고리즘이 가장 좋은 결과를 도출하고 있으므로, 제안 알고리즘의 성능이 우수한 것을 알 수 있다.

표 1 시뮬레이션 결과(배분전력량)

Table 1 Simulation Results(kW)

목표전력량	Greedy	시간제약하의 최적 알고리즘	시간제약하의 제안 알고리즘
110,001	110,300	110,310	<b>110,010</b>
120,001	120,300	<b>120,010</b>	<b>120,010</b>
130,001	130,300	130,110	<b>130,010</b>
140,001	150,000	140,160	<b>140,010</b>
150,001	150,300	150,520	<b>150,010</b>
160,001	170,000	160,110	<b>160,010</b>
170,001	170,300	170,060	<b>170,010</b>
180,001	180,300	180,170	<b>180,010</b>
190,001	190,300	190,410	<b>190,010</b>
200,001	200,300	200,240	<b>200,010</b>
210,001	210,300	210,050	<b>210,010</b>
220,001	230,000	220,550	<b>220,010</b>
230,001	230,300	230,670	<b>230,010</b>
240,001	240,300	240,040	<b>240,010</b>
250,001	250,300	250,800	<b>250,010</b>
260,001	260,300	260,610	<b>260,010</b>
270,001	270,300	270,800	<b>270,010</b>
280,001	290,000	280,900	<b>280,010</b>
290,001	290,300	290,130	<b>290,010</b>
300,001	300,300	300,980	<b>300,010</b>

### 5. 결론 및 고찰

본 논문에서는 Polynomial 시간안에 수행이 가능한 새로운 부하배분 알고리즘을 제시하였다. 주어진 수행시간 내에서 새로운 알고리즘이 기존 알고리즘보다 좋은 성능을 가지고 있음을 시뮬레이션을 통하여 검증하였다. 최근 고유가로 인하여 효율적인 수요관리의 필요성이 점차로 커지고 있으며, 멀지 않은 시간에 직접부하제어는 입찰 시장에 진입할 것으로 보인다. 최적의 부하배분 알고리즘은 직접부하제어 시스템의 효율적인 운영에 도움이 될 수 있을 것으로 기대된다.

### 참 고 문 헌

- [1] 김형중, et al., "직접부하제어 시스템의 운영 및 구성방안", 2003년도 대한전기학회 하계학술대회논문집, pp. 627-629, 2003.
- [2] 김진호, et al., "직접부하제어자원의 활용방안에 대한 연구", 2003년도 대한전기학회 하계학술대회논문집, pp. 606-608, 2003.
- [3] 정구형, et al., "부하관리사업자의 비상시 부하제어량 배분 알고리즘 개발", 2003년도 대한전기학회 하계학술대회논문집, pp. 633-635, 2003.
- [4] 에너지관리공단, 2004년 에너지관리공단 직접부하제어 운영 기준 및 수용가 직접부하제어시스템 기술규격서, 2004.
- [5] 손학식, et al., "직접부하제어 시스템의 효율적인 제어 절차 방안 연구", 2004년도 대한전기학회 하계학술대회논문집, pp. 642-644, 2004.

## 저 자 소 개



#### 김 정 욱 (金 政 郁)

1993년 2월 한국과학기술원 전기전자공학(공학박사), 1989년 2월 한국과학기술원 전기전자공학(공학석사), 2007년 3월-현재 호서대학교 벤처전문대학원 교수, 정보관리기술사/정보통신기술사/정보시스템감리사,

<관심분야> : 부하제어, 신재생에너지